NUMERICAL MODELING FOR THE FORMATION MECHANISM OF 3D TOPOGRAHPY ON MICROBIAL MAT SURFACES

An Undergraduate Research Scholars Thesis

by

HARSH JAY PATEL

Submitted to Honors and Undergraduate Research
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Research Advisor: Dr. Michael Tice

May 2014

Major: Civil Engineering

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Numerical Modeling for the formation mechanism of 3D topography on Microbial Mat Surfaces.
(May 2014)

Harsh Jay Patel
Zachry Department of Civil Engineering
Texas A&M University

Research Advisor: Dr. Michael Tice
Department of Geology and Geophysics

Microbial mats are layered, surface-attached communities of microorganism that live under stagnant or moving bodies of water. Active precipitations of minerals often occur around these spatially arranged communities, forming rough, conical or domal surfaces. These surfaces are considered Earth's oldest and most robust fossil signatures of life whenever microfossils are unattainable. Despite decades of investigation, the formation mechanisms of these shapes remain vaguely explained. It was hypothesized though, that nutrient limitation coupled with fluid motion may play a key role as a physical control. Under this model, competitions of nutrients were setup among growing microbial communities, which later evolve into specially arranged, 3D mats. However, this hypothesis seems to require an initial condition, a template of early topographical randomness for the physical mechanisms to kick in. This initial surface was observed in laboratory grown mats, but its physical role was never investigated. For this research, an innovative modeling approach was employed that focuses on the interface growth of the microbial mat surfaces using a combined stochastic and deterministic approach. A range of different initial conditions were simulated to evaluate the 3D topography evolution. This method directly assists the experimental work on mats growth, and allows a more robust test of the possible biological mechanisms that exist in forming various surfaces in the rock record, thus offering a better interpretation.

DEDICATION

I dedicate this research to my parents. Without their support this research opportunity would not have been possible.

ACKNOWLEDGEMENT

I would like to thank **Prof. Michael Tice (Texas A&M University)** and the **Mr. Jian Gong (Texas A&M University)** for their tremendous support. Their teaching has been paramount to the production of this thesis and my improvement as a student and person.

I would also like to thank **Prof. Eric Verracchia (University of Lausanne, Switzerland)** and **Dr. Damien Chappatte (University of Lausanne, Switzerland)** for providing the necessary software programs and its source code.

NOMENCLATURE

KPZ	Kardar-Parisi-Zhang
DLA	Diffusion Limited Aggregation
CA	Cellular Automata
CO_2	Carbon dioxide
FDM	Finite Difference Method
FEM	Finite Element Method
PDEs	Partial Differential Equations

CHAPTER I

INTRODUCTION

1.1 Introduction to Microbial Mats

Microbial life collectively transformed the surface conditions of our planet and has continuously been doing so. Understanding its record is key to interpreting environments of the past and to guide us into managing our ecosystems in the future. Most microbial communities reside on some surfaces that are subject to certain physical forces while various metabolic tasks are performed. Biological activities such as growth and motility impact the spatial community structure, leading to styles of spatial heterogeneity that could be the key to start as well as to continuously steer the morphogenesis of microbialites -- the time capsules of biological community states recorded as mineralized shapes. A complex suite of processes, given physical, chemical or biological may influence the production and erosion of these structures. Despite our best efforts, still many fundamental processes have remained covert to our understanding, which hinders our ability to fully comprehend the records.

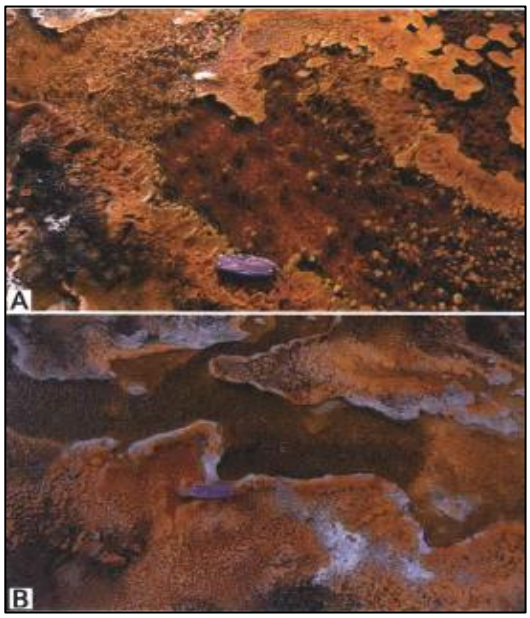


Figure 1: (A) Conical Microbial Mats in still side pools of hot springs in Yellowstone National Park (Petroff, 2011). (B) Flat Microbial Mats in fast moving sections of the stream (Petroff, 2011).

1.2 Previous Models

Previous models already exists to explain the morphology of microbial mats. Three of the most common of the already existing models are the Kardar-Parisi-Zhang non-linear model, the Diffusion Limited Aggregation-Cellular Automata model, and the Reaction-Diffusion model.

1.2.1 The Kardar-Parisi-Zhang (KPZ) Model

The non-linear equation of Kardar-Parisi-Zhang, which simulated the profile evolution of a growing interface, is used for modelization of various growth problems (Kardar, Parisi, & Zhang, 1986). A modified version of KPZ equation was proposed by Grotzinger and Rothman in 1996 in order to question the biogenic origin of some stromatolites (Grotzinger & Rothman, 1996). The KPZ equation includes parameters such as surface-normal accretion, surface tension, and noise that allow knowledge regarding the morphogenesis of much simpler stromatolite forms (Dupraz, Pattisina, & Verrecchia, 2006). The modified non-linear KPZ model is deterministic in approach and produces iterative vertical growth of regular, smooth and compact laminae (Dupraz, Pattisina, & Verrecchia, 2006). However, this model is very unstable under certain simulation conditions and can only simulate growth of massive stromatolites in a bulk (Dupraz, Pattisina, & Verrecchia, 2006). The lateral growth limitation of this model does not allow production of branching forms of stromatolites that are often observed in Proterozoic geologic time period (Dupraz, Pattisina, & Verrecchia, 2006).

1.2.2 The Diffusion Limited Aggregation-Cellular Automata (DLA-CA) Model

The Diffusion Limited Aggregation-Cellular Automata is a stochastic model, used to investigate stromatolite morphological space (Dupraz, Pattisina, & Verrecchia, 2006). It is formulated using the Diffusion Limited Aggregation method described by Witten and Sander in 1981 and 1983 (Witten & Sander, 1981). DLA is the formation of aggregates of particles undergoing Brownian motion (random walk) until contact is made an aggregate cluster of a

substrate (Witten & Sander, 1981). The diffusion method described by the DLA model is different from the traditional diffusion method, which assumes a normal flow system where all particles move in approximately the same direction (Dupraz, Pattisina, & Verrecchia, 2006). The shape of the aggregate cluster in the DLA model is controlled by the ability of particles to reach the cluster (Dupraz, Pattisina, & Verrecchia, 2006). In simulations with initially low particle concentration, the aggregation process is performed one particle at a time and is often very slow (Dupraz, Pattisina, & Verrecchia, 2006).

In the DLA-CA model, the surface roughness of the stromatolite progressively increases with high points of the cluster (Dupraz, Pattisina, & Verrecchia, 2006). Logically, high points of the aggregate cluster catches more particles. Thus, not all parts of the build-up (cluster) have equal growth probabilities (Dupraz, Pattisina, & Verrecchia, 2006). The use of the DLA model also introduces the notion of self-similarity, which releases the model from the scale restraints (Dupraz, Pattisina, & Verrecchia, 2006). This means that different phenomenon can thus be modeled at various scales with the same tool (Dupraz, Pattisina, & Verrecchia, 2006).

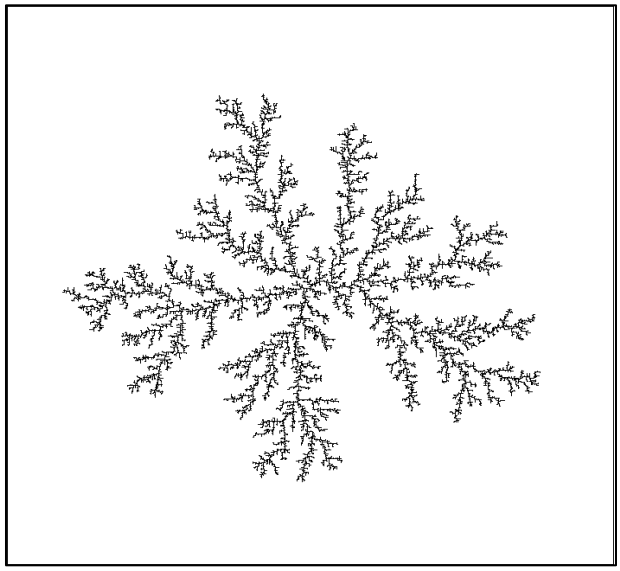


Figure 2: The result of a DLA simulation in 2D as described by Witten and Sanders (Witten & Sander, 1981). A single particle is set as the anchor particle (fixed permanently) at the center of the geometric space, and new particles are then generated one at time and move stochastically until they reach the aggregate cluster and stick to it (Chappatte, 2010).

In the DLA-CA model, the DLA part simulates the main extrinsic growth factors of the stromatolites whereas the CA are the systems of cells showing simple local interactions that are able to describe complex global behaviors. In a Cellular Automata, time and space are discrete variables, i.e. the space is subdivided into cells considered as individual entities that have a defined state at a time t. The progression from state of time $t = i$ to another $t = t + i$ is described by transition rules and the state of the cell at the time t. The growth resulting from the Cellular Automata is called 'local', because this particular growth does not depend on the global structure of the build-up, but on simple transition rules, locally determined as a function of the state of one cell and its neighbors.

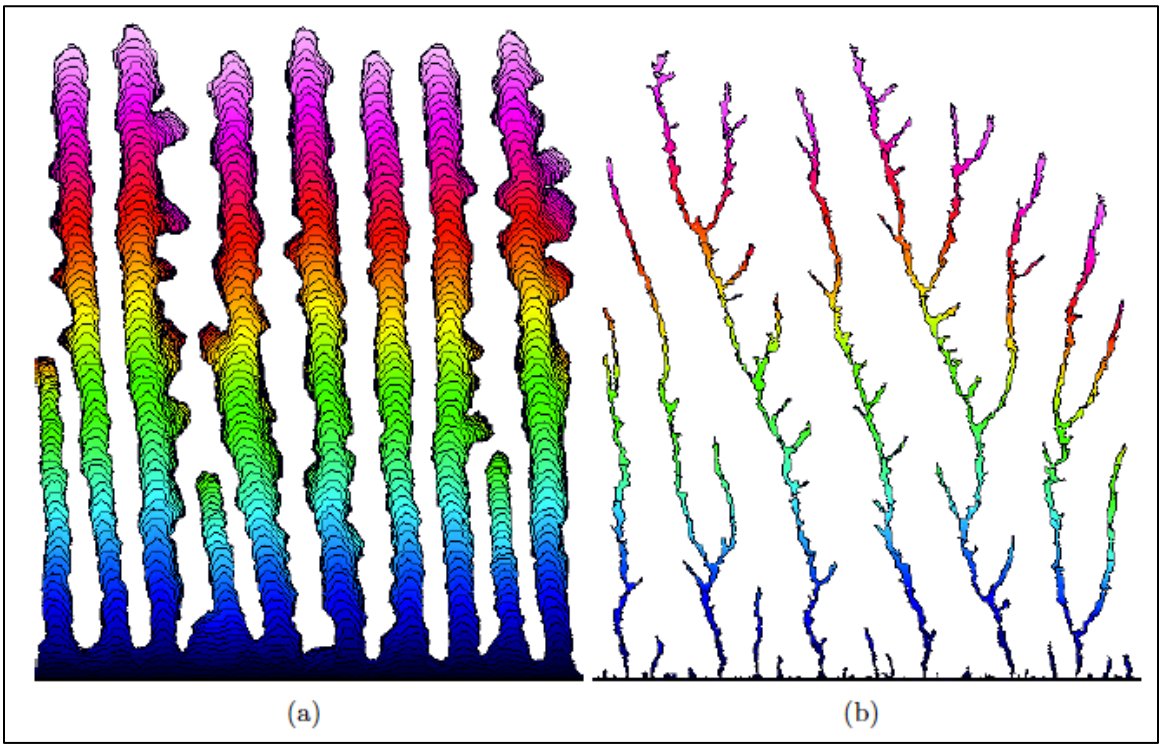


Figure 3: Two examples of simulation of stromatolite growth using DLA-CA method in 2D. Image obtained through DLA-CA Growth software (Chappatte, 2010).

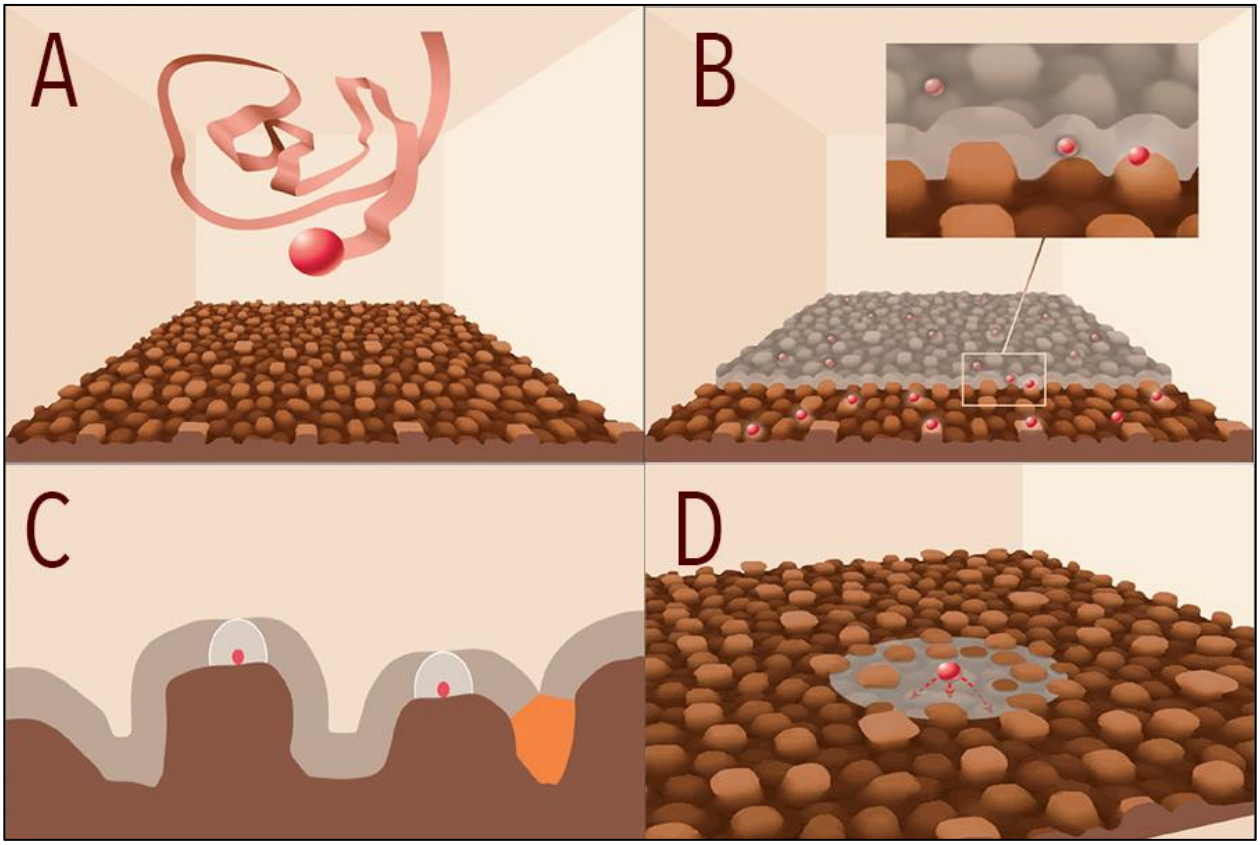


Figure 4: Illustration of the DLA-CA method in 3D. (A) Particles fall one at time from top of the space until they reach the aggregate cluster. (B) Particles get attracted to the aggregate cluster when they enter the attraction zone (greyish laminate). (C) Slice of the aggregate (brown) with attraction zone (grey) and two particles (red) and their stability zone (white). (D) When a particle reaches the aggregate, every position at the lower level and at a distance larger or equal to the sedimentation distance is checked. If one of the positions is free, the particle moves to that position or stays at its current location. Image obtained through DLA-CA Growth software (Chappatte, 2010).

1.2.3 The Reaction-Diffusion Model

Simple reaction-diffusion models predict that limitation by nutrient diffusions sets up competitions for sites where diffusive pathways are shortest within microbial communities (Petroff, 2011). Under certain environmental conditions, existing surface relief of mats can be amplified, resulting regularly spaced surface roughness features. Although some of the special case microbialites were found to form in relatively quiet waters, most microbial mats experience some fluid flow such that their overlying diffusive boundary layers are modified by currents or waves and may not necessarily scale with depth (Chang, Gilbert, Eliashberg, & Keasling, 2003).

1.3 Research Objectives

To date most existing numerical models on stromatolite formation are 2D models. However, the structures of microbial mats are innately three-dimensional. All turbulent flows are 3D structures and actions as well (Gong, 2011). Most of the microbial mat models to date ignore the effect of turbulence and its effect on enhanced diffusive transport process and erosion (Gong, 2011). It is the goal of this research to eventually produce an efficient scheme to model mats growth in 3D. This work lays the foundation for future modeling work that can be easily adopted and expended. Realizing that modeling turbulence alone in 3D itself requires significant computation resources, and often introduces problems in model construction (Gong, 2011), it is also my hope to take advantage of the latest algorithms in computation fluid dynamics and make progresses computationally. This work may be useful for understanding environments in the past and how evolution in microbial communities progressed over time. In particular, this project employs computational techniques, and focuses on the physical processes in fluids such as advection and diffusion to understand their roles in connection with biological growth on surfaces.

CHAPTER II

METHODS

2.1 Introduction to Deterministic vs. Stochastic Solute Diffusion Model

A deterministic solute diffusion model has no stochastic elements and the input and output relation of the model is conclusively determined. A deterministic model includes both a dynamic model, and a static model. Compared to a deterministic solute diffusion model, a stochastic diffusion model has one or more stochastic elements. Concentration diffusion system formulated stochastically cannot be solved analytically due to the complexity of the problem. In the case of simulating a stochastic model, a random number is normally generated by some method to execute trial.

Simulation done by the deterministic model is often considered one of the specific instances of a simulation by the stochastic model. Given that there are no random elements in the deterministic model, deterministic simulation can only be done once (usually with a pen and paper if problem is not too complex). On the other hand, in stochastic model, once the value has been derived by using a random number, the simulation does not suffer from deterministic simulation in simultaneous stages.

2.2 Deterministic Solute Diffusion Model

Deterministic model of the concentration diffusion equation (Eq. 1) is described by a differential equation. By solving this differential equation iteratively, realistic physical processes can be simulated. Well-defined linear equations generally have a unique solution. Whereas non-linear equations, when solved numerically, may have multiple solutions.

$$\frac{\partial c}{\partial t} = -D\nabla^2 c \ (Eq. 1)$$

Where:

- c is the variable of interest, CO_2 concentration
- D is the diffusivity constant for chemical compound (CO_2) that is being diffused
- ∇ is the gradient (Laplace Operator) (∇^2 for quasi-1D solute diffusion is $\frac{\partial^2 c}{\partial^2 x}$, and ∇^2 for quasi-2D solute diffusion is $\frac{\partial^2 c}{\partial^2 x} + \frac{\partial^2 c}{\partial^2 y}$)

The solute diffusion equation can be solved using numerical methods after they are properly discretized. Some of the most common numerical methods used in geo-modeling are finite difference method (FDM), finite element method (FEM), and path simulation. For this research, only FDM is considered, and is elaborated on.

2.2.1 Finite Difference Method (FDM)

The deterministic solute diffusion model for this research follows time series of steady state events. This means that concentration at all spatial values in the discretized grid (x and y, z was not modeled for this research) is measured for "real-time" temporal variations.

The principle of Finite Difference Method for solute diffusion equation is close to the other numerical schemes used to solve ordinary differential equations. It approximates the difference operator by replacing the derivatives in the equation using differential quotients. For quasi-1D solute diffusion equation, the domain is partitioned in x-space and in time t and the approximations of the solution are then computed at the space or time points. For quasi-2D solute diffusion equation, the domain is partitioned in x, y-space and in time t and the approximations of the solution are then computed at the space or time points. Two of the most common FDMs are the explicit scheme, and the implicit scheme. Only the explicit scheme is discussed in this thesis.

Explicit Scheme

The explicit scheme for solute diffusion equation calculates the state of the state of solute diffusion system at a later time from the state of the system at the current time.

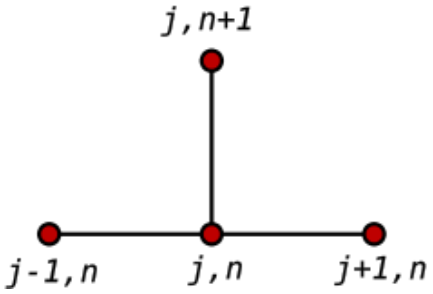


Figure 5: Illustration of the explicit scheme. Explicit scheme, also known as *Forward in Time* scheme, uses current state of the system at $t = i$ to calculate the state of the system at $t = i + 1$. (Slingerland & Kump, 2011)

A. Quasi-1D Solute Diffusion Equation

The quasi-1D solute diffusion equation in reality involves two variables, x, and t. For the explicit quasi-1D solute diffusion equation (Eq. 2), concentration variation is measured at different values of x for varying time values.

$$\frac{c_{i+1}-c_i}{t_{i+1}-t_i} = -D\,\frac{c_{i+1}-2c_i+c_{i-1}}{\Delta x^2} \quad (Eq.2)$$

Where:

- c is the variable of interest, concentration
- D is the diffusivity constant for chemical compound (CO_2) that is being diffused
- t is the time
- Δx is the spatial interval in horizontal direction

In order to deterministically solve the quasi-1D concentration equation, boundary conditions are specified before solving the problem. The system is considered steady state, and the effect of constant flow on solute diffusion is ignored. The concentration at the upper boundary condition is set to $c = 100$. The concentration at the lower boundary condition, or the

mat surface is set to $c = 0$. The left and right boundaries are set as reflective boundaries to simulate constant flow and avoid out of bound errors. Also, the spatial variable x is divided into 100 intervals, so $\Delta x = 1$.

B. Quasi-2D Solute Diffusion Equation with Mats Growth

The quasi-2D solute diffusion equation in reality involves three variables, $x, y,$ and t. x describes the horizontal positioning of each microbialite in a 2D microbial mat, whereas y describes the height of each microbialite. Due to computational limitations, same height was assigned to each microbialite using a sinusoidal equation. This phenomenon, discussed above, is described by a master Equation 3b (derived from Equation 3a) which describes the growth of the microbial mat due to solute diffusion.

$$\frac{\partial \vec{m}}{\partial t} = \left(\frac{\partial m_x}{\partial t} + \frac{\partial m_y}{\partial t}\right) \ (Eq. 3a)$$

$$\frac{\partial m_x}{\partial t} = \alpha D \frac{\partial c}{\partial x}, and \ \frac{\partial m_y}{\partial t} = \alpha D \frac{\partial c}{\partial y}$$

$$\frac{\partial \vec{m}}{\partial t} = \alpha D \vec{\nabla} c \ (Eq. 3b)$$

Where:

- c is the CO_2 concentration
- D is the diffusivity constant for chemical compound (CO_2) that is being diffused
- α is a constant
- m is the variable of interest (microbialite growth vertically and horizontally with respect to time)

Unlike the quasi-1D solute diffusion equation, which merely appears in the form of a number line, the quasi-2D solute diffusion equation forms a Cartesian grid system. Also, for the explicit quasi-2D solute diffusion equation (Eq. 4), concentration variation is measured at different values of x and y for varying time values.

$$\frac{c_{i+1}-c_i}{t_{i+1}-t_i} = -D\left(\frac{c_{i+1}-2c_i+c_{i-1}}{\Delta x^2} + \frac{c_{i+1}-2c_i+c_{i-1}}{\Delta y^2}\right) \ (Eq.4)$$

Where:

- c is the variable of interest, concentration
- D is the diffusivity constant for chemical compound (CO_2) that is being diffused
- t is the time
- Δx is the spatial interval in horizontal direction
- Δy is the spatial interval in vertical direction

The boundary conditions for the quasi-2D solute diffusion equation are very similar to that of quasi-1D solute diffusion equation. It is considered steady state, and the effect of constant flow on solute diffusion is ignored. The concentrations at the upper and lower boundaries are set to $c = 100$ and $c = 0$, respectively. The left and right boundaries are set as reflective boundaries to simulate constant flow and avoid out of bound errors. Also, the spatial variable x $(length)$ and y $(height)$ are divided into 100 intervals, so Δx and Δy equals to 1.

2.3 Stochastic Solute Diffusion Model

The stochastic model of the solute diffusion equation developed from the previous work done by Damien Chappatte and Jaap Kaandorp (Chappatte, 2010) (Kaandorp, Lowe, Frenkel, & Sloot, 1996). Chappatte's and Kaandorp's stochastic diffusion model is derived from DLA model introduced by Witten and Sanders in 1983.

As discussed in Chapter I, DLA cluster grows by accumulating particles that move purely by diffusion. The movement of the accumulating particles is affected by both flow and diffusion. The relative importance of these two affects on the concentration transport is characterized using the Péclet number, Pe (Eq. 5) (Kaandorp, Lowe, Frenkel, & Sloot, 1996).

$$Pe = \frac{ul}{D} \ (Eq. 5)$$

Where:

- Pe is the Péclet number
- D is the diffusivity constant for chemical compound (CO_2) that is being diffused
- u is the mean flow velocity
- l is characteristic length

A low value of the Péclet number indicates that accumulating particles move mainly by the diffusion process, whereas a high value of the Péclet number indicates that their motion is primarily dominated by the flow (Kaandorp, Lowe, Frenkel, & Sloot, 1996).

2.3.1 Lattice Boltzmann Scheme

The flow around the aggregate cluster is modeled using the Lattice Boltzmann equation. The lattice method is the most adequate for this particular scenario because of three principal reasons:

1. The Lattice Boltzmann scheme is stable when dealing with flows around complex objects
2. It is easier to model concentration particles as “tagged” gas particles
3. The Lattice Boltzmann scheme is suitable for parallel computations

The state of the Lattice Boltzmann fluid is specified by the average number of accumulating particles with velocity c_i, at each link, which is denoted as $n_i(r,t)$. The time evolution of the distribution function n_i is described by the discretized analog of the original Boltzmann equation. The probability that a particle moves with a velocity c_i after a collision is calculated using $\frac{n_i(r,t)}{\rho\,(r,t)}$, where $\rho\,(r)$ is the total number of particles at the node. By introducing tagged particles that are identical to the particles constituting the Lattice Boltzmann fluid, but have a probability of $\frac{\Delta}{\rho\,(r,t)}$ that they will remain at the site, it is also possible to vary the diffusion coefficient of the tracer. The time evolution of $P\,(r,t)$ for tracer particles, which are completely absorbed at the solid-fluid interface, is calculated using Equation 6.

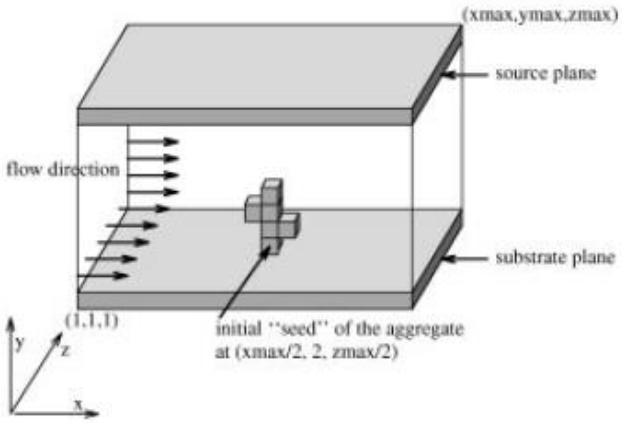


Figure 6: Illustration of the basic construction of the aggregate. The fluid flow in the illustration is modeled using the Lattice Boltzmann scheme (Kaandorp, Lowe, Frenkel, & Sloot, 1996).

$$P(r,t+1) = \sum_{i \neq i_b}^{b} \frac{\left[n_i(r-c_i,t) - \frac{\Delta}{b}\right] P(r-c_i,t)}{\rho(r-c_i,t)} + \Delta \frac{P(r,t)}{\rho(r,t)} \quad (Eq.6)$$

(Kaandorp, Lowe, Frenkel, & Sloot, 1996)

Where:

- i_b is boundary link, connecting a node in the fluid to solid (aggregate or the particle)
- b number of discrete velocities

In order to stochastically solve the stochastic solute diffusion equation (Eq. 6), initial and boundary conditions are specified before solving the problem. Periodic boundary conditions are applied for the fluid, at the faces of simulation (Kaandorp, Lowe, Frenkel, & Sloot, 1996). At the solid-fluid interface a "no-slip" boundary condition is applied, but an absorbing boundary condition is applied to the tracer particles (Kaandorp, Lowe, Frenkel, & Sloot, 1996). Antiperiodic boundary conditions are applied on the faces of the simulation box (Kaandorp, Lowe, Frenkel, & Sloot, 1996).

2.4 Establishing Initial Conditions

2.4.1 Control Initial Condition

The control initial condition is a flat microbial mat surface with no microbialite growth. It is essentially a 3D printed plastic layer of several centimeters. Having a control initial condition allows to analyze how results differ from pseudo-stochastic initial conditions. It also allows to formulate or check the mathematical relationship between diffusion and growth morphology if one exists in nature.

2.4.2 Pseudo-Stochastic Initial Condition

A pseudo-stochastic point generator is used to mimic "pseudo-stochastic" placement of conical/domal microbialites on the hypothetical 5 inches by 5 inches microbial mat flat surface. The purpose of this is to physically simulate stochasticity regarding where the microbialite grows.

The algorithm for the pseudo-stochastic point generator is developed using the *Random Point Grid Placement* method. For example, the program initiates by generating a 5 inches by 5 inches square grid with 1 inch by 1-inch squares. The 5 inches by 5 inches square illustrates a flat microbial mat surface, whereas, squares created by the grid illustrate sub-compartment where each microbialite will be placed. After the grid is set up, a random point (with values between 0

and 1) is generated with x and y values, and is "placed" or "plotted" into the square generated by points (0,0), (1,0), (0,1), and (1,1).

Once a randomly generated point is placed into the first square of the square grid, another point is generated and placed into the second square. The process continue until all the squares have a random point. Also, the program is restricted from placing more than one random point in any given square. This allows to control the stochasticity, and monitor important characteristics such as zone of influence, and microbialite spacing. After the program has successfully executed, the output is used as the initial condition for the differential equations.

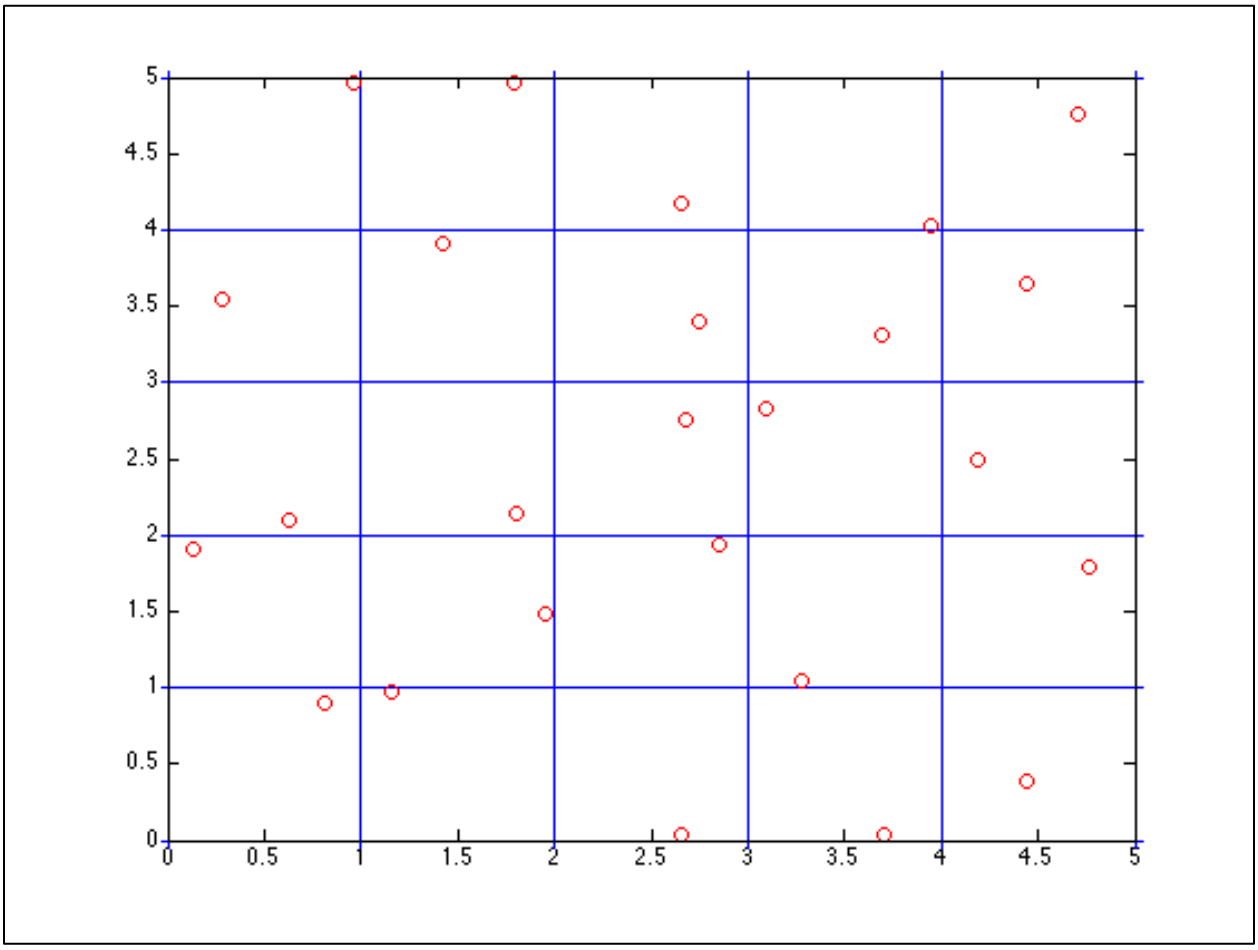


Figure 7: Sample output from the Pseudo-Stochastic Point Generator program. As stated, only one random point is placed in each sub-compartment. This program allows us to determine initial conditions for the Lattice Boltzmann equation.

CHAPTER III

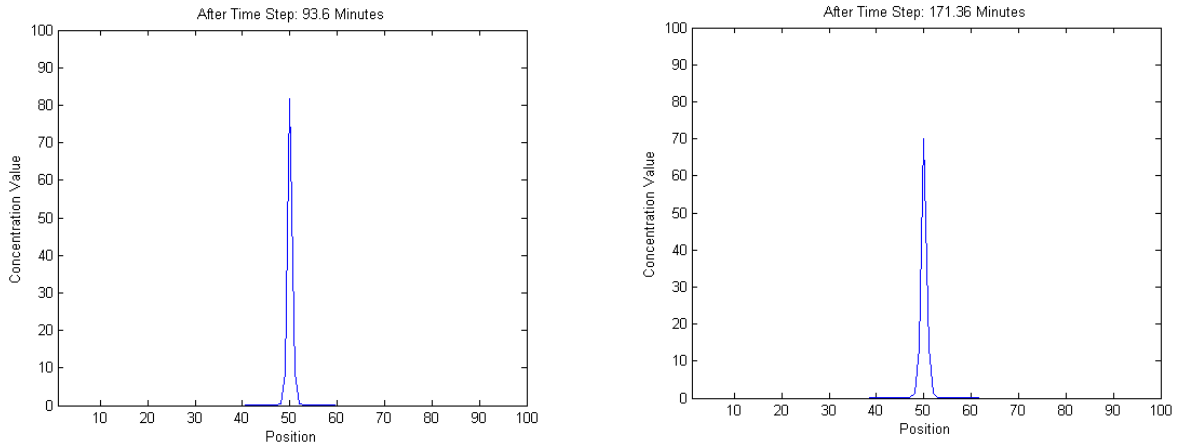
RESULTS

3.1 Deterministic Solute Diffusion Model

The deterministic solute diffusion model is solved using PDEs that describes the solute diffusion for x and y spatial variables for a range of temporal variations (time, t). The resulting equations for quasi-1D and quasi-2D solute diffusion, are then solved using FDM explicit scheme in the MATLAB© program.

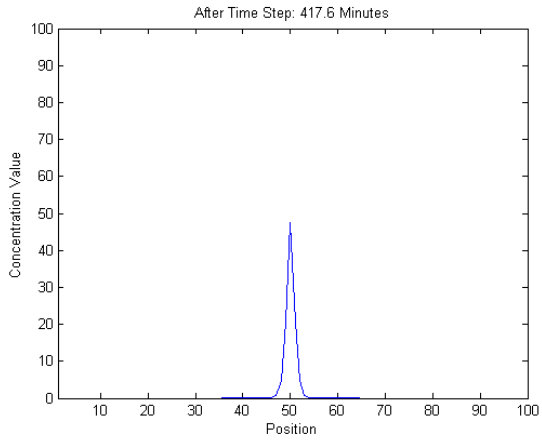
Figure 8 illustrates how the diffusion of the CO_2 concentration progresses in one-dimensional (x) steady fluid flow with respect to time. As seen in the series of images in Figure 8, passive diffusion process is very slow in time. Based on the observations from simulation, 100 CO_2 concentration takes more than a week (image not shown) to completely diffuse in water. This correlates with the reasoning that why nutrient transport solely through diffusion, is one of the inhibiting factors for the growth of the microbial mats.

Figure 9 illustrates how the diffusion of the CO_2 concentration affects the growth of the microbial mats (height) with respect to time. It is a quasi-2D solute diffusion model because both the concentration and the height of microbialite cones are measured at different time values. As seen in the series of images in Figure 9, passive diffusion process is very slow in time. The initial shape, given by a sinusoidal function $hcone(xbloxks) = 40 * abs(sin(0.05 * xblocks))$, and starting height of the microbial mat is specified before executing the program.

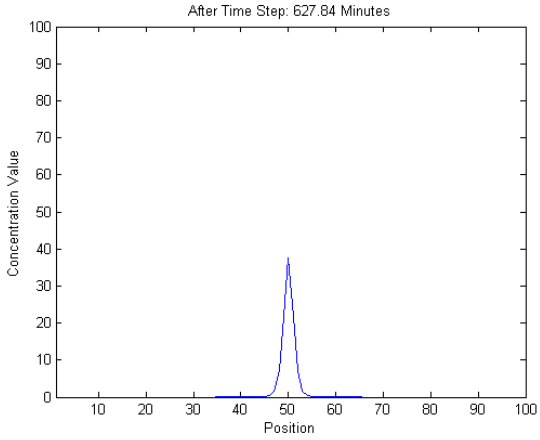


(A)

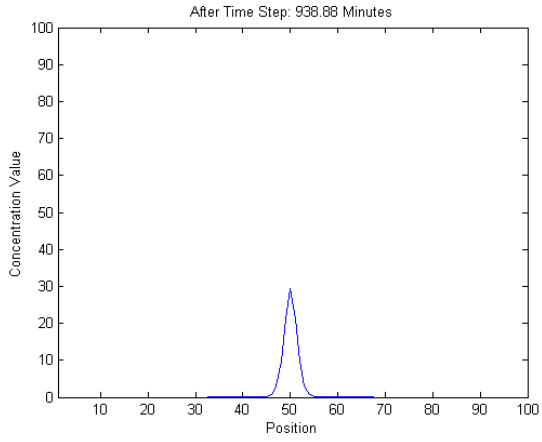
(B)



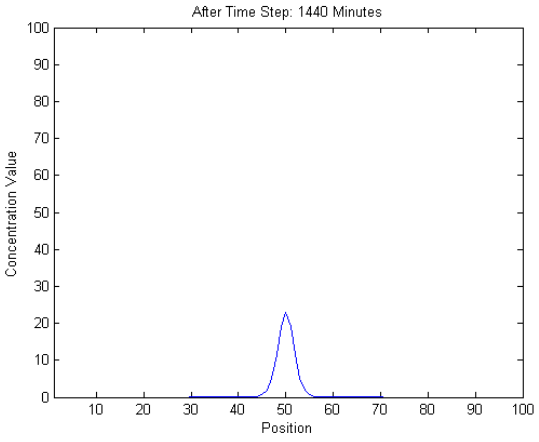
(C)



(D)



(E)



(F)

Figure 8: Quasi-1D solute diffusion model in MATLAB. Images A through F depict solute diffusion in one-dimensional steady fluid flow with respect to time. Nutrient transport through diffusion process can very slow and an inhibiting factor for microbial mats. It takes more than a week for 100 CO_2 concentration to completely diffuse in water (Image is not shown).

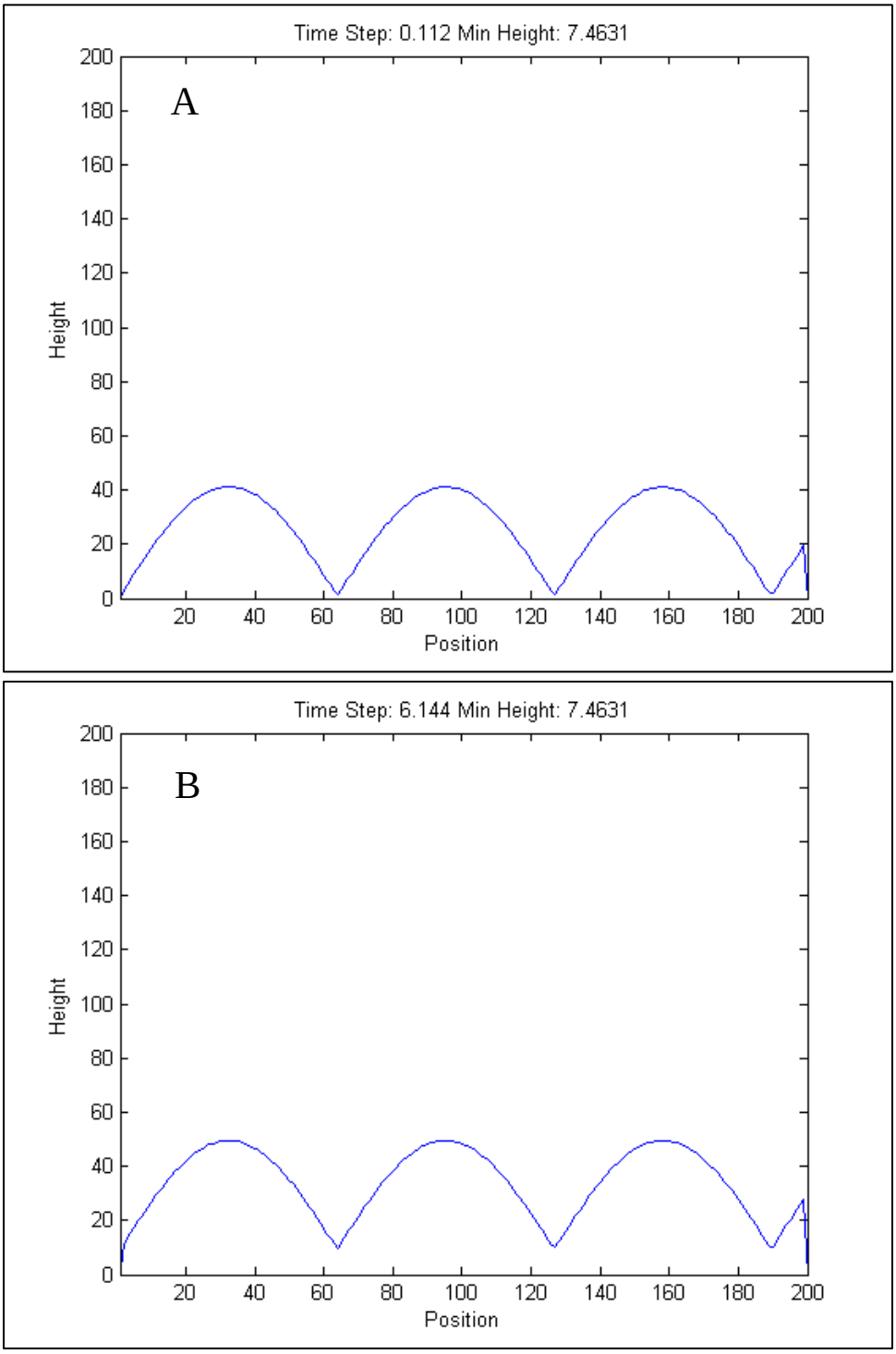


Figure 9: Quasi-2D solute diffusion model with mats growth in MATLAB. Images A and B depict solute diffusion and change in microbial mat height with respect to time. As time progresses, the height of the microbial mats change accordingly. This means that at certain point in time, in absence of a flow, lack of nutrient transport through diffusion will inhibit and stop the vertical growth of the microbial mats.

3.2 Stochastic Solute Diffusion Model

Unlike the deterministic solute diffusion equation, the stochastic equation is modeled in quasi-3D. This means in reality it has $x, y, z,$ and t variables. Special software package called DLA 3D EXT (Chappatte, 2010) is used to produce necessary outputs. DLA 3D EXT program uses the principle of the Lattice Boltzmann scheme discussed earlier in Chapter III. The program is divided into two parts, a Simulator and a Visualizer. The Simulator part of the program requires a user input such as the initial and the boundary conditions, whereas the Visualizer part is used to see the output result.

Figure 10 illustrates a time series of microbialite growth at 5%, 10%, 20%, 55%, 75%, 90%, and 100% growth using stochastic model implemented in DLA 3D EXT program. As seen in Figure 11, a sinusoidal surface was used for the microbialite growth to compare it with results from quasi-2D deterministic solute diffusion model (Figure 9). The cross-sections illustrated in Figure 10, has the same underlying connotation as Figure 9 except that they are derived using two different models. Since the stochastic form of the solute diffusion equation is not a dynamic model, different values of time are represented using color variation in Figure 10. The purple color at bottom indicates that this laminae or layer was deposited earlier in time as compared to the blue and yellow laminae at the top. Also, the columnar growth and laminaes as seen in Figure 12 (B), agrees with the stromatolite and microbialite growth seen in the field Figure 12 (A).

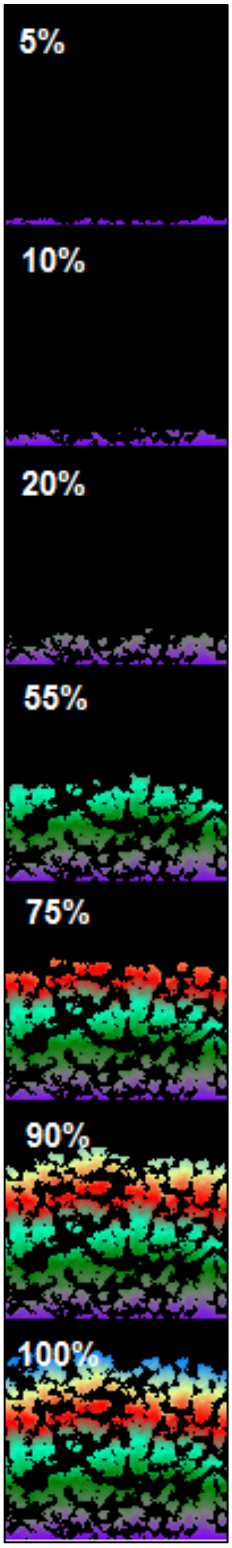


Figure 10: Time series of microbialite/stromatolite growth at different growth percentages using stochastic model implemented in the DLA 3D EXT program.

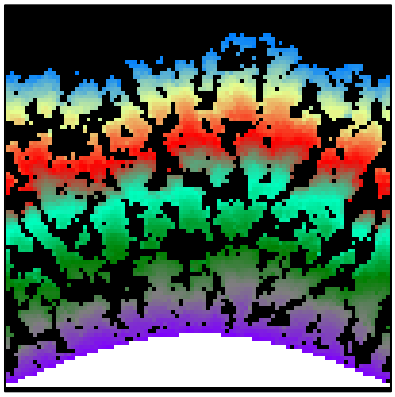


Figure 11: Illustration of sinusoidal base surface implemented in DLA 3D EXT program. It allows comparing the results from 2D stochastic model with the results from quasi-2D deterministic model (Figure 9).

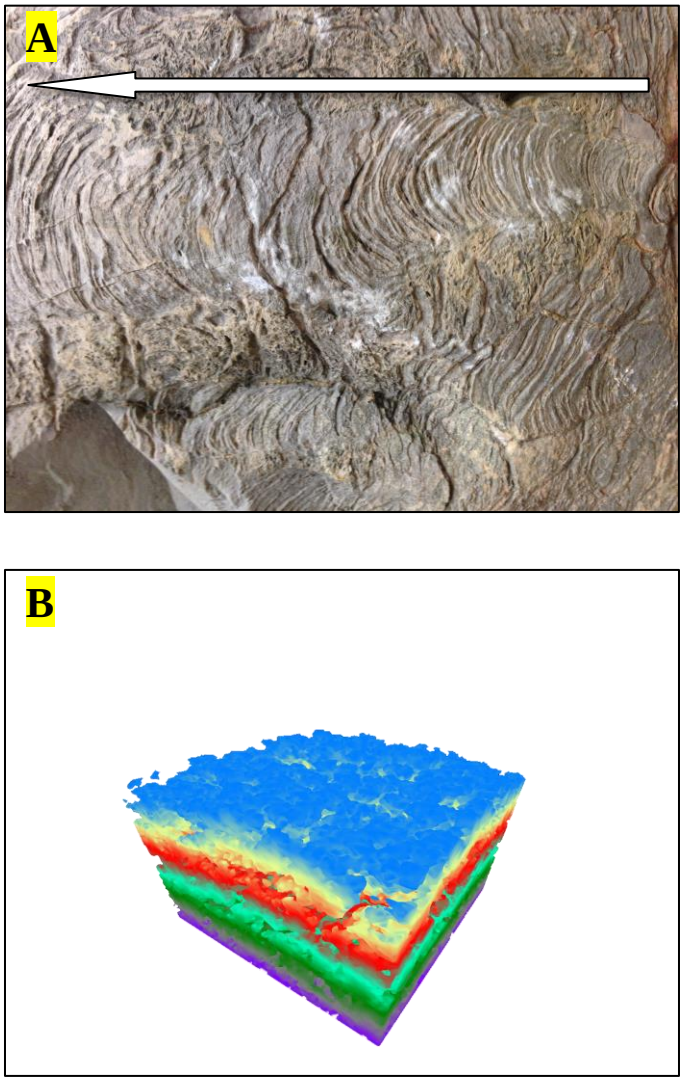


Figure 12: The columnar growth and the resulting laminaes as seen in stochastic model (B), agree with the stromatolite and microbialite growth seen in the field (A).

CHAPTER IV

CONCLUSION

After doing the post analysis, it can be concluded that the results from this research seem promising, but contradicting. Concentration diffusion does affect the growth morphology of the microbial mats. However, this conclusion is only true when both, deterministic and stochastic approaches are considered. A true mathematical relationship would not require assumptions except for the initial and the boundary conditions.

The FDM explicit scheme, deterministic in approach, is only stable for low values of input parameters. This suggests that the applying FDM limits problem constraints that can be used for a given situation. Implicit scheme, although not used in this research, is expected to have similar results to that explicit scheme.

The Lattice Boltzmann scheme, and the stochastic approach are indeed very stable, however the problem formulation requires several more assumptions other than the initial and the boundary conditions. This contradicts with the modeling purpose in the first place, because true relationship between solute diffusion and growth morphology cannot be predicted using artificial inputs. This makes the system pseudo-stochastic instead of fully stochastic as desired.

Due to time limitations, several methods initially proposed, were never implemented. For example, it is expected that successful application of FEM with Crank-Nicolson scheme might make the deterministic approach more stable. Given that this research is incomplete in its conclusion, better technique can definitely be applied to formulate a more robust model that will give a true, universal, mathematical relationship.

REFERENCES

Chang, I., Gilbert, E. S., Eliashberg, N., & Keasling, J. D. (2003, June 26). A three-dimensional, stochastic simulation of biofilm growth and transport-related factors that affect structure. *Microbiology*(149), 2859-2871.

Chappatte, D. (2010). *Simulation of stromatolite growth using diffusion-limited aggregation and a discrete model of biogeochemical exchanges in a microbial mat.* PhD Thesis, University of Lausanne, Faculty of Geosciences, Lausanne.

Dupraz, C., Pattisina, R., & Verrecchia, E. (2006). Translation of energy into morphology: Simulation of stromatolite morphospace using a stochastic model. *Sedimentary Geology, 185*(3-4), 185-203.

Gong, J. (2011). *Finite Element Method for Biofilm Simulation.* Texas A&M University, Department of Geology and Geophysics.

Grotzinger, P., & Rothman, D. (1996). An abiotic model for stromatolite morphogenesis. *Physical Review Letters, 383*, 423-425.

Kaandorp, J. A., Lowe, C. P., Frenkel, D., & Sloot, P. M. (1996, September 9). Effect of Nutrient Diffusion and Flow on Coral Morphology. *Physical Review Letter, 77*(11).

Kardar, M., Parisi, G., & Zhang, Y. (1986). Dynamic scaling of growing interface. *Physical Review Letters*(56), 889-892.

Petroff, A. P. (2011). *Streams, stromatolites and the geometry of growth.* PhD Thesis, Massachusetts Institute of Technology, Department of Earth Atmospheric and Planetary Sciences.

Slingerland, R., & Kump, L. (2011). *Mathematical Modeling of Earth's Dynamical Systems - A Primer.* Princeton, NJ, USA: Princeton University Press.

Witten, T. A., & Sander, L. M. (1981). Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical Review Letters*(47), 1400-1403.

APPENDIX

Appendix I: quais1D Solute Diffusion MATLAB code

```
% General_Diffusion_1D.m
% 1-Dimensional Diffusion Model
% Harsh Jay Patel; Mr. Jian Gong; Dr. Michael Tice
% Geo-Surface Modeling Lab
% Department of Geology and Geophysics, Texas A&M University
% ----------------------------------------------------------------------------------------------------------------------------------------%

total_distance = 1;                                 % length of x-axis (position)
cellblocks = 100;                                   % sub-divisions of x-axis
delta_x = total_distance / cellblocks;              % length of each sub-divisons

total_simulation_time = 5*24*60*60;                 % total simulation time in seconds
timesteps = 1000;                                   % number of time intervals to perform
delta_t = total_simulation_time / timesteps;        % size of each time interval (incremental timesteps)

D_CO2 = 1.92E-9;                                    % diffusivity constant for CO2 (m^2/s)

Values = zeros(cellblocks,timesteps);               % creates a 100 x 1000 matrix with zeros
MaxConc = 100;                                      % maximum inital concentration that is to be diffused
Values(cellblocks/2,1) = MaxConc;                   % resets the concentration at the midpoint of the cellblocks array to 100

for i = 1:timesteps                                 % performs a looped calculation for total of number of timesteps

  for x = 2:cellblocks-1

     % calculate concetration for each cell with incrementing timesteps except for the 1st or the 100th cell
     Values(x,i+1) = Values(x,i) + D_CO2*(delta_t/(delta_x)^2)*(Values(x+1,i)-2*(Values(x,i))+Values(x-1,i));

     % calculate concetration for the 1st cell with incrementing timesteps
     Values(1,i+1) = Values(1,i) + D_CO2*(delta_t/(delta_x)^2)*(2*Values(2,i)-2*(Values(1,i)));

     % calculate concetration for the 100th cell with incrementing timesteps
     Values(cellblocks,i+1) = Values(cellblocks,i) + D_CO2*(delta_t/(delta_x)^2)*(2*Values(cellblocks-1,i)-...
     2*(Values(cellblocks,i)));

  end

  plot(Values(:,i))                                 % plot the values
  axis([1 cellblocks 0 MaxConc])                    % set the minima and maxima on axes

  % label the axes and the title
  xlabel('Position')
  ylabel('Concentration Value')
  title(['After Time Step: ' num2str(i*delta_t/60) ' Minutes'])

  drawnow                                           % make Matlab display the graph

end
```

Appendix II: quais2D Solute Diffusion MATLAB code

```
% Mats_Growth_Diffusion_2D.m
% 2-Dimensional Diffusion Model with Mats Growth
% Harsh Jay Patel; Mr. Jian Gong; Dr. Michael Tice
% Geo-Surface Modeling Lab
% Department of Geology and Geophysics, Texas A&M University
% ----------------------------------------------------------------------------------------------------------------------------------------%

x_distance = 1;                                 % x-distance in meters
xblocks = 200;                                  % number of cells for the x-distance array
delta_x = x_distance / xblocks;

y_distance = 1;                                 % y-distance in meters
yblocks = 200;                                  % number of cells for the y-distance array
delta_y = y_distance / yblocks;

total_simulation_time = 60*60;                  % seconds (for 24 hours)
timesteps = 15000;                              % Number of time intervals to perform
delta_t = total_simulation_time / timesteps;

D_CO2 = 2.02E-5;                                % diffusivity constant of CO2 (m^2/s)
C_VAPOR = 2;
alpha = 50;                                     % alpha constant
D_CO2_num = D_CO2*(delta_t/(delta_x)^2);

Stability_Condition_Met = delta_t <= delta_x*delta_y/(2*D_CO2)          % stability condition

%% Initialize Blank Data Structure
C = zeros(xblocks,yblocks,timesteps);
hcones = zeros(xblocks,timesteps);
dhcones = zeros(xblocks,1);

%% Initial Conditions
C(:,yblocks,1) = C_VAPOR;
hcones(:,1) = 40*abs(sin(0.05*(0:1:xblocks-1)));
x = 1;                                          % intializes x to 1 (x is used in the loop)
for t = 1:timesteps-1                           % performs a looped calculation for total of number of timesteps
   % Boundary Conditions
   C(:,yblocks,t) = C_VAPOR;                    % top boundary
   C(1,:,t) = C(2,:,t);                         % left boundary
   C(xblocks,:,t) = C(xblocks-1,:,t);           % right boundary
   for i = 2:xblocks-1
      dhcones(i) = round(hcones(i,t));
      C(i,dhcones(i)+1,t) = 0;                  % bottom Mat Surface as Boundary
   end
   for y = min(dhcones)+2:yblocks-1             % performs a looped calculation for concentration diffusion in the y-axis
      for x = 2:xblocks-1                       % performs a looped calculation for concentration diffusion in the x-axis
         % Equations
         C(x,y,t+1) = C(x,y,t) + D_CO2_num*( (C(x+1,y,t)-2*(C(x,y,t))+C(x-1,y,t)) + (C(x,y+1,t)-2*(C(x,y,t))+C(x,y-1,t)));
         b = (C(x,y+1,t)-C(x,y,t))/delta_y;
         a = (C(x+1,y,t)-C(x,y,t))/delta_x;
         hcones(x,t+1) = hcones(x,t) + (b/(sqrt(a^2+b^2)))*delta_t*alpha*D_CO2*(b + a);
      end
   end

  plot(hcones(:,t))                             % plot the values
  axis([1 xblocks 0 200])                       % set the minima and maxima on axes
  xlabel('Position')                            % labels the x-axis of the plot
  ylabel('Height')                              % labels the y-axis of the plot
  % titles the plot
  title(['Time Step: ' num2str(t*delta_t/60) ' Min' ' Height: ' num2str(max(hcones(:,t))-hcones(20,t))])
  drawnow                                       % make Matlab display the plot
end
```

Appendix III: quais2D Solute Diffusion MATLAB code

```
% pseudo_rand_point_gen.m
% Pseudo-Stochastic Point Generator
% Harsh Jay Patel; Mr. Jian Gong; Dr. Michael Tice
% Geo-Surface Modeling Lab
% Department of Geology and Geophysics, Texas A&M University
% ----------------------------------------------------------------------------------------------------------------------------------------%

num = 6;
grid_matrix=zeros(num,num,2);
random_matrix=zeros(num-1,num-1,2);

for i=1:num
   for j=1:num
      grid_matrix(i,j,1) = i;
      grid_matrix(i,j,2) = j;
   end
end

for i=1:num-1
   for j=1:num-1
      random_matrix(i,j,1) = grid_matrix(i,j,1) + rand();
      random_matrix(i,j,2) = grid_matrix(i,j,2) + rand();
   end
end

% shifting for plotting
grid_matrix = grid_matrix - ones(num,num,2);
random_matrix = random_matrix - ones(num-1,num-1,2);

plot(grid_matrix(:,:,1),grid_matrix(:,:,2),'-+b');
hold on;
plot(grid_matrix(:,:,2),grid_matrix(:,:,1),'-b');
plot(random_matrix(:,:,1),random_matrix(:,:,2),'ro');
```